



DECUS

PROGRAM LIBRARY

DECUS NO.	12-133
TITLE	MINT - MULTIPLE PRECISION INTEGER ARITHMETIC SUBROUTINE
AUTHOR	Larry Davis Washington University, St. Louis, Missouri
COMPANY	Submitted by: Robert Hassinger Liberty Mutual Research Center Hopkinton, Massachusetts
DATE	October 9, 1972
SOURCE LANGUAGE	SABR

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

MINT - MULTIPLE PRECISION INTEGER ARITHMETIC SUBROUTINE

DECUS Program Library Write-up

DECUS NO. 8-631

ABSTRACT

ARITHMETIC AND INPUT-OUTPUT SUBROUTINES ARE PROVIDED FOR
MULTIPLE PRECISION INTEGERS.

USAGE

THESE SUBROUTINES MAY BE CALLED FROM EITHER FORTRAN OR
SABR. THE FILE MINT.RL MUST BE LOADED WITH THE
PROGRAM OR PLACED IN LIB8.

STORAGE OF INTEGERS

***** ** *****

INTEGERS ARE STORED IN SUCCESSIVE MEMORY LOCATIONS.
AN N-WORD INTEGER IS CONSIDERED AS A $12 \times N$ BIT TWO'S
COMPLEMENT NUMBER. THE LEAST SIGNIFICANT WORD IS STORED FIRST.

EXAMPLE:

USING 3-WORD INTEGERS, THE DECIMAL NUMBER 16,786,131
(OCTAL 000100020003) IS STORED AS

0003

0002

0001

THE DECIMAL NUMBER -2 IS STORED AS

7776

7777

7777

THE PACKAGE USES A PSEUDO-ACCUMULATOR. THE NUMBER OF WORDS IN THE ACCUMULATOR MAY BE VARIED UNDER PROGRAM CONTROL BY THE MPREC ROUTINE UP TO THE MAXIMUM OF WORDS. THIS MAXIMUM IS SET AT ASSEMBLY TIME OF MINT. IN THE PRESENT PACKAGE, THE MAXIMUM IS 11 WORD INTEGERS. TO CHANGE THIS SEE THE NOTES AT THE END OF THIS DESCRIPTION. ALL ARITHMETIC OPERATIONS LEAVE THEIR RESULTS IN THE PSEUDO-ACCUMULATOR. THE ACCUMULATOR MAY BE STORED WITH THE MSTO ROINE.

STORAGE ALLOCATION FOR INTEGERS

***** ***** *** *****

IN FORTRAN, SPACE MAY BE ALLOCATED BY USING INTEGER ARRAYS.
IN SABR, SPACE MAY BE ALLOCATED BY THE BLOCK PSEUDO-OPERATION.

SUMMARY OF ROUTINES

***** ** *****

NOTE: IN THE FOLLOWING, IA REFERS TO A MULTIPLE PRECISION INTEGER, AND I AND J REFER TO SINGLE-PRECISION INTEGERS.

A. MPREC

FORT: CALL MPREC(I)

SABR: CALL 1, MPREC
ARG I

THIS ROUTINE WILL CHANGE THE SYSTEM TO USE I-WORD INTEGERS.

B. MCLR

FORT: CALL MCLR

SABR: CALL 0, MCLR

THIS ROUTINE WILL CLEAR ALL WORDS OF THE PSEUDO-ACCUMULATOR.

C. MADD

FORT: CALL MADD(IA)

SABR: CALL 1, MADD
ARG IA

THIS WILL ADD THE MULTIPLE-PRECISION INTEGER AT IA TO THE PSEUDO-ACCUMULATOR.

D. MSUB

FORT: CALL MSUB(IA)

SABR: CALL 1, MSUB
ARG IA

THIS WILL SUBTRACT THE INTEGER AT IA FROM THE PSEUDO-ACCUMULATOR.

E. MCOM

FORT: CALL MCOM

SABR: CALL 0, MCOM

THIS WILL GIVE THE TWO'S COMPLEMENT OF THE PSEUDO-ACCUMULATOR.

F. MABS

FORT: CALL MABS

SABR: CALL 0, MABS

THIS WILL GIVE THE ABSOLUTE VALUE OF THE PSEUDO-ACCUMULATOR.

G. MSCR

FORT: CALL MSCR

SABR: CALL 0, MSCR

THIS WILL SHIFT THE PSEUDO-ACCUMULATOR RIGHT 1 BIT, FILLING WITH A ZERO.

H. MSCL

FORT: CALL MSCL

SABR: CALL 0, MSCL

THIS WILL SHIFT THE PSEUDO-ACCUMULATOR LEFT 1 BIT, FILLING WITH A ZERO.

I. MSGN

FORT: I=MSGN(0)

SABR: CALL 1,MSGN
ARG 0
DCA I

THIS IS A FUNCTION. ITS ARGUMENT IS IGNORED. IT RETURNS AN INTEGER WHOSE VALUE IS:

- 1 IF THE PSEUDO-ACCUMULATOR IS NEGATIVE
- 0 IF THE PSEUDO-ACCUMULATOR IS ZERO
- 1 IF THE PSEUDO-ACCUMULATOR IS POSITIVE

J. MPMY

FORT: CALL MPMY(IA)

SABR: CALL 1,MPMY
ARG IA

THIS WILL MULTIPLY THE VALUE IN THE PSEUDO-ACCUMULATOR BY THE INTEGER AT IA. OVERFLOW IS IGNORED.

K. MDIV

FORT: CALL MDIV(IA)

SABR: CALL 1,MDIV
ARG IA

THIS WILL DIVIDE THE VALUE IN THE PSEUDO-ACCUMULATOR BY THE INTEGER AT IA.

L. MREM

FORT: CALL MREM(IA)

SABR: CALL 1,MREM
ARG IA

THIS WILL STORE IN THE INTEGER IA THE REMAINDER FROM THE PREVIOUS CALL TO MDIV. THIS MUST BE CALLED IMMEDIATELY AFTER MDIV OR THE RESULTS CANNOT BE GUARANTEED, SINCE OTHER ROUTINES MAY DESTROY THIS VALUE.

M. MSTO

FORT: CALL MSTO(IA)

SABR: CALL 1, MSTO
ARG IA

THIS WILL STORE THE PSEUDO-ACCUMULATOR AT IA.

N. MBAS

FORT: CALL MBAS(I)

SABR: CALL 1, MBAS
ARG I

THIS WILL CHANGE THE BASE FOR INPUT AND OUTPUT TO I.
NOTE: I MUST BE BETWEEN 2 AND 47. HOWEVER, NO CHECK IS MADE
FOR I OUTSIDE THIS RANGE. THE DIGITS WILL BE ACCORDING TO
8-BIT ASCII CODE, STARTING AT 0.
THEREFORE, 10 IS REPRESENTED BY :
ETC.

O. MIN

FORT: CALL MIN(I)

SABR: CALL 1, MIN

ARG I

THIS WILL READ AN INTEGER INTO THE PSEUDO-ACCUMULATOR IN THE CURRENT BASE FROM FORTRAN DEVICE NUMBER I. THE INTEGER MUST BE TERMINATED BY A CARRIAGE RETURN. IF AN ERROR IS MADE, TYPE RUBOUT, AND A BACKSLASH AND CARRIAGE RETURN AND LINE FEED WILL BE TYPED ON DEVICE 1. THEN THE WHOLE INTEGER MUST BE RETYPED. NEGATIVE NUMBERS MUST HAVE A "-" SIGN. BLANKS ARE IGNORED.

P. MOUT

FORT: CALL MOUT(I,J)

SABR: CALL 2,MOUT
ARG I
ARG J

THIS WILL OUTPUT THE LOW-ORDER J DIGITS OF THE INTEGER IN THE PSEUDO-ACCUMULATOR ON DEVICE I IN THE CURRENT BASE. FOR EXAMPLE, IF J=5, IT WOULD OUTPUT IN AN I5 FORMAT. THE NUMBER IS PADDED ON THE LEFT WITH BLANKS. THE ACCUMULATOR IS CLEARED AFTER OUTPUT.

ERROR MESSAGES

A. MDVZ ERROR AT LOC XXXX

CAUSE: AN ATTEMPT WAS MADE TO DIVIDE BY ZERO.

ACTION TAKEN: THE LARGEST POSITIVE INTEGER IS RETURNED,
AND EXECUTION CONTINUES.

B. MPRC ERROR AT LOC XXXX

CAUSE: A PRECISION WAS SPECIFIED WHICH WOULD REQUIRE
MORE WORDS IN THE PSEUDO-ACCUMULATOR THAN ALLOCATED.

ACTION TAKEN: EXECUTION TERMINATES AND CONTROL RETURNS
TO THE PS/8 MONITOR.

INCREASING THE MAXIMUM PRECISION

***** **

CURRENTLY, THE SYSTEM ALLOWS A MAXIMUM OF 11 WORDS PER INTEGER. THE ROUTINES AND ACCUMULATOR FIT INTO 6 PAGES WHEN LOADED.

TO INCREASE THE NUMBER OF WORDS ALLOCATED, CHANGE MWORDA, AC1, AC2, AC3, AND OUTTB IN THE SOURCE, AND REASSEMBLE. THE BLOCK XX STATEMENTS MAY HAVE TO BE SPLIT UP INTO MORE THAN ONE BLOCK PSEUDO-OP TO KEEP SABR FROM WASTING SPACE AT THE END OF A PAGE.

NUMBER OF CORE PAGES REQUIRED	MAXIMUM NUMBER OF WORDS/INTEGER	MAXIMUM NUMBER OF DECIMAL DIGITS PER INTEGER(EXCLUDING SIGN)
6	11	39
7	29	104
8	47	169
9	65	234
10	83	299
30	443	1599

NOTE: SINCE THIS IS DONE IN SOFTWARE, THE LARGER THE NUMBER SPECIFIED AS AN ARGUMENT TO MPREC, THE LONGER THE TIME REQUIRED FOR THE OPERATION.

THE FOLLOWING CHANGES WERE MADE TO THE SOURCE TO ALLOW INTEGERS OF 450 WORDS/INTEGER.

MWORDA, -702 /THIS MUST BE -(WORDA)

BASE, 12

THE FOLLOWING ARE THE PSEUDO-ACCUMULATORS.

[illegible]